Inteligencia Artificial 2017-2018

Práctica 4: Torneo de Heurísticas

GRUPO: 2213

CELIA SAN GREGORIO MORENO

ÁLVARO MARTÍNEZ MORALES

Índice

Pregunta C1

A lo largo de la práctica 4 se han implementado varias heurísticas diferentes y, aunque cada tenga su propio aproximamiento al problema de la cuantificación del tablero, muchos son el resultado de variaciones de la implementación de una heurística previa.

Las distintas versiones entregadas de cada una de las heurísticas simplemente cambiaban la ponderación de los componentes del tablero, nunca la aproximación de la cuantificación en si.

Por último, queda destacar que aquí figuran las heurísticas que han tenido un verdadero planteamiento y trabajo de elaboración detrás; algunas de las heurísticas subidas durante los primeros días no se ven reflejadas en este documento puesto que eran simples pruebas del resultado de heurísticas sin ningún tipo de complejidad.

Petacabras

Este fue nuestra primera implementación heurística, y su objetivo era el de crear un bot que jugara defensivamente teniendo en cuenta la diferencia de puntuación frente al jugador contrario.

Así, lo primero que tuvimos en cuenta era, en el caso de que la partida hubiera terminado en el estado evaluado, si era él el ganador, caso en el que la heurística evaluaría el estado como (a efectos prácticos) infinito, y, de ser él el perdedor, lo evaluaría como menos infinito.

Seguidamente, comprobaba si el estado le permitía saltar turno, en cuyo caso evaluaría un tablero como un valor de infinito ligeramente menor al de partida ganada (osease, a efectos prácticos, infinito, pero manteniendo una prevalencia sobre la el estado del victoria.)

Por último, si no se daban ninguna de esas situaciones en el estado evaluado, evaluaría, en primera instancia (y con una valor superior) la diferencia entre puntos del Kalaha propio sobre el contrario, dando prioridad a estados en los que la puntuación definitiva del jugador fuera mayor sobre la del contrario.

En segunda instancia, evaluaría **negativamente** mediante una función exponencial la cantidad de semillas en cada uno de los hoyos propios (excluyendo el kalaha). Esto se ha hecho de manera que el bot procurase acumular la menor cantidad posible de semillas en cada uno de sus hoyos, para dejar el menor número posible de semillas expuestas a robos. Precisamente por eso se ha hecho uso de una función exponencial frente a una más simple de carácter lineal; la cantidad de semillas a ser robadas que expones no son las mismas si tienes 2 semillas en 3 hoyos que 3 semillas en 2 hoyos; en un caso te pueden robar, como máximo, 2 y en el otro 3, aunque, en el fondo, haya la misma cantidad de semillas en nuestros hoyos.

Por último cabe destacar que sobre la mitad del periodo de prácticas decidimos incluir el caso de empate como caso de derrota (previamente estaba dispuesto al contrario) para no lanzarnos a él atribuyéndole la misma evaluación que una victoria.

Las distintas versiones tendieron hacia una ponderación heurística que ponderaba más negativamente acumular semillas en los últimos hoyos (más a la derecha) lo que daba prioridad a realizar jugadas en ese area de nuestro campo.

Petabuses

El objetivo de esta heurística es crear un bot defensivo, que valore positivamente algunos factores y muy negativamente otros.

En primer lugar, la heurística evalúa si el juego ha terminado. En caso de derrota o empate, ponderará muy negativamente sobre el tablero (se escogió un número muy grande para simular el valor infinito). En caso de victoria, ponderará muy positivamente.

Si el juego no ha terminado, realiza una suma con estos parámetros:

a) La resta de las fichas propias en el Kalaha con las fichas del contrario en su Kalaha. Cuantas más fichas tenga el jugador actual en el Kalaha (siendo ese número de fichas mayor que el del contrario), la heurística ponderará más favorablemente esa diferencia positiva.

   Si en la disposición del tablero el jugador actual tiene menos fichas en su Kalaha que el contrario, la heurística ponderará negativamente.

   Con esta resta pretendemos priorizar la acumulación de fichas en el Kalaha propio, ya que no pueden ser capturadas y permiten asegurar la puntuación.

b) La suma conjunta de unas operaciones de multiplicación. Estas multiplicaciones se realizan con el hoyo propio, el hoyo opuesto y un valor numérico arbitrario.

   El objetivo de las multiplicaciones es ponderar un hoyo propio teniendo en cuenta el número de fichas del hoyo opuesto. Cuantas más fichas tenga el jugador contrario en sus hoyos, más posibilidades hay de capturar estas fichas si en los hoyos propios hay 0 fichas.

Send_Nodes

La heurística de este bot es una modificación de la función de evaluación del bot Petabuses. Las ponderaciones sobre el fin del juego, la posibilidad de pasar turno y el número de fichas en el Kalaha propio y el del contrario son idénticas (salvo ligeros cambios en los valores numéricos).

La particularidad de esta heurística se encuentra a la hora de ponderar los hoyos. Si en el tablero actual algún hoyo del contrario tiene 0 fichas, la función de evaluación pondera negativamente, ya que no hay posibilidad de captura directamente (y el jugador contrario podría capturar las fichas del actual).

En cambio, si el hoyo del contrario tiene 1 ficha o más, se sirve de las ponderaciones utilizadas previamente en el bot Petabuses para determinar cuán recomendable resulta mover una ficha en función del número de fichas en los hoyos del contrario.

Minuano

Esta heurística surgió a partir de un intento por evaluar el tablero en función de este parámetro: la posibilidad que tenía el bot de conseguir un turno más en función del número de fichas en sus hoyos.

De nuevo, nos basamos en la heurística del bot Petabuses. Ello puede apreciarse en la evaluación del fin de partida, la situación en la que el bot puede jugar una vez más y el número de fichas en los Kalahas propio y contrario.

A todos estos parámetros se añadió la diferencia de puntuación entre el jugador actual y en contrario. La heurística ponderaría positivamente cuanta más puntuación tuviese el jugador actual respecto al contrario, y negativamente en caso contrario.

Además, si en alguno de los hoyos propios había fichas suficientes para sembrar hasta el Kalaha propio, la heurística ponderaba muy positivamente. Sino, ponderaba cuán recomendable era escoger un hoyo para realizar la siembra con valores arbitrarios.

Cabe destacar que este bot no obtuvo muy buena posición en el torneo, ya que en la mayoría de los casos escogía posiblemente la ponderación muy positiva si se cumplía la condición de poder pasar turno en alguno de los hoyos.

Y seguramente esa decisión le impidiera considerar otras estrategias más interesantes.

Jetstream-Sam

La función de evaluación de este bot surgió a raíz de un intento por configurar una heurística distinta a las anteriores, que valorase negativamente el hecho de pasar turno (porque no sabíamos con certeza si el jugador actual era el que pasaba turno o no) y el número de fichas en el hoyo actual en función del hoyo opuesto.

En cuanto a las ponderaciones sobre el número de fichas, tuvimos en cuenta dos factores:

a) Que el hoyo contrario tuviese 0 fichas.
b) Que el hoyo actual tuviese un número grande de fichas (5 o más).

A la diferencia del número de fichas en los Kalahas, calculado de igual forma que en los bots anteriores, restábamos unas multiplicaciones de los hoyos propios y los opuestos. Estas multiplicaciones tenían la estructura:

$$hoyo\_propio * \frac{120}{hoyo\_contrario + 1}$$

Cuantas más fichas tuviese el hoyo propio y cuantas menos fichas tuviese el hoyo contrario, la operación devolvería un valor negativo más grande. Y, por tanto, sería menos recomendable mover y acumular fichas en los hoyos propios si en los opuestos hay muy pocas fichas o incluso cero.

Este bot tampoco obtuvo una buena posición en el torneo. Posiblemente, al no aplicar la regla de la captura por avaricia (si en un hoyo hay 5 fichas o más), sus valoraciones sobre el tablero no fueron relevantes.

Lyssa

Esta variación de **Petacabras** incluía, sobre lo anteriormente expuesto en la aproximación de dicha heurística, una evaluación sobre el caso de salto de turno, el cual ya no ponderaría simplemente un infinito por debajo únicamente del estado de victoria, si no que se evaluaría en dicho caso sumando a nuestra puntuación la mayor cantidad de semillas en un hoyo contrario expuesta a robo (osease, con 0 semillas en la casilla contraria del campo propio), de tal forma que calcularíamos las ganancias potenciales del salto, y nos evitaríamos saltar en situaciones en las que no saltando, pero tomando otra jugada que derivaría en un robo, ganaríamos muchas más semillas que en el caso de salto.

Este bot derrotaba con creces a todo lo que habíamos implementado hasta la fecha. Desafortunadamente, también, sin explicación aparente, tenía una misteriosa tendencia a abandonar en situaciones fortuitas pero deterministas (osease, sin ningún tipo de explicación, pero siempre contra determinados bots) por un probable bug en la implementación del juego, lo que le ha impedido alcanzar resultados mucho mejores que su predecesor, pues estos abandonos cuentan como derrotas para nuestro bot. De ahí su nombre, *Lyssa*, la diosa de griega de la ira, la furia y el frenesí (y, en este caso, de ragequit).

Maniae

Esta última heurística fue, al igual que *Lyssa,* basada en **Petacabras**, pero, frente a los sofisticados cambios de la previa, esta, símplemente, dejó de contemplar la posibilidad de salto en su heurística.

Soprendentemente, no quedaba en absoluto lejos de su hermana, aunque en esta nunca nos encontramos casos del bug.

Pregunta C2

Probablemente esto se deba a que, dado que la implementación del Negamax considera que el oponente tiene tu misma heurística, el jugador Bueno se reservaría de tomar ciertas buenas jugadas en base a consideraciones a las que no conseguía alcanzar su hermano jugador Regular, que seguía yendo hacia las jugadas que consideraba mejores sin esa clase de miramientos.

Eso nos pasó, por ejemplo, con la mayoría de nuestras heurísticas que juagaban contra last-opt; este nunca ganaba porque jugase mejor que las nuestras, si no porque forzaba a jugar de una manera defensiva a nuestras heurísticas, de forma que estas tendían a jugar distribuyendo fichas por su campo y alejándolas de las posiciones libres (y robables) del campo de last-opt, aunque este en realidad nunca tendría ningún interés en robar fichas mediante esos hoyos vacíos. Sin embargo, nuestras heurísticas no sabían eso, considerando que jugaban contra ellas mismas, acababan por perder contra algo infinitamente más simple...

Por tanto, esta solución reside en pequeños cambios que provocasen al bot un enfoque de juego menos defensivo y más directo, más, incluso, símple. Sin embargo esta sería una mala estrategia a la larga, puesto que esto lo llevaría a tener peores resultados contra heurísticas más avanzadas.